



KOREAN PATENT ABSTRACTS(KR)

Document Code:B1

(11) Publication No.1001442950000

(44) Publication.Date. 19980416

(21) Application No.1019940034105

(22) Application Date. 19941214

(51) IPC Code:

H04N 7/24

(71) Applicant:

DAEWOO ELECTRONICS CO., LTD.

(72) Inventor:

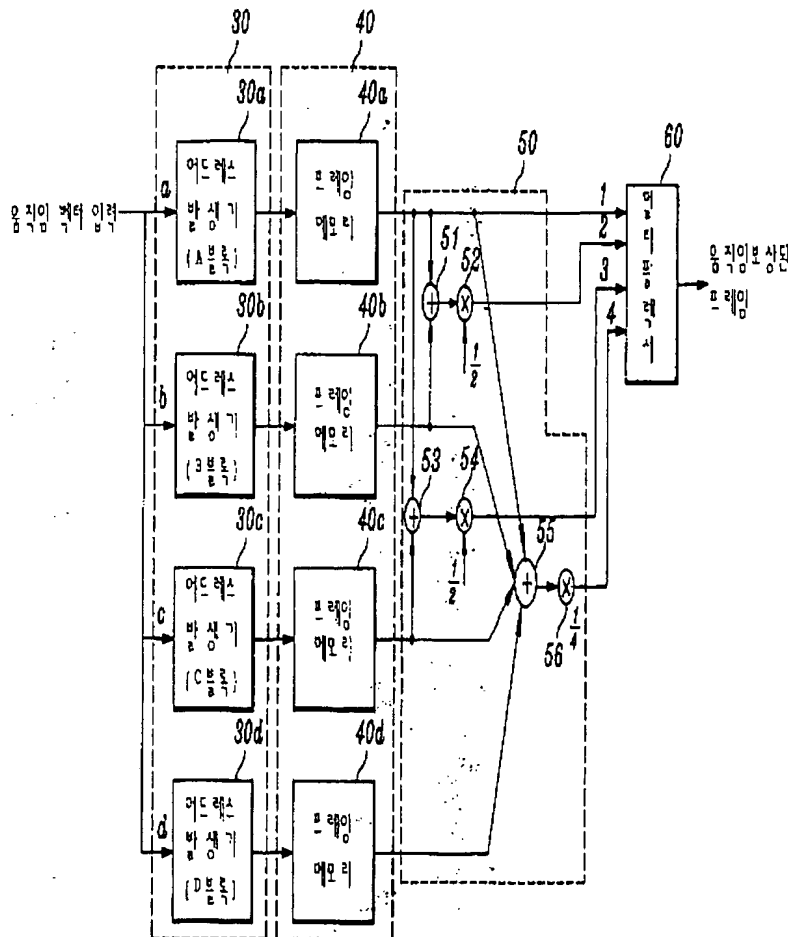
JUNG, HAE MOOK

(30) Priority:

(54) Title of Invention

DEVICE AND METHOD FOR OVERLAPPED MOVEMENT COMPENSATION

Representative drawing



(57) Abstract:

PURPOSE: A device and a method for the overlapped movement compensation are provided to prevent the blocking occurred when encoding images by using a block matching algorithm.

CONSTITUTION: A device is composed of : an address generation part(30) ; a memory part (40) ; an output switching part(50) ; and a multiplexer(60). The address generation part consists of four address generation units (30a,30b,30c,30d) generating each address according to the movement vector values of each divided block. The memory part consists of four frame memories (40a,40b,40c,40d) corresponding to each address generation unit.

The output switching part consists of three adders(51,53,55) and three subtractors(52,54,56). And the multiplexer is

for the selection of the overlapped parts due to being overlapped of the divided
each block A,B,C,D.

COPYRIGHT 2000 KIPO

if display of image is failed, press (F5)

(19) 대한민국특허청(KR) (12) 공개특허공보(A)

(51) Int. Cl.⁶
H04N 7/13

(11) 공개번호 특1996-0028436
(43) 공개일자 1996년07월22일

(21) 출원번호 : 특1994-0034105
(22) 출원일자 : 1994년12월14일
(71) 출원인 : 대우전자 주식회사 배순훈
서울특별시 중구 남대문로 5가 541번지 (우 : 100-095)
(72) 발명자 : 정해묵
서울특별시 마포구 공덕 2동 현대아파트 102-1305
(74) 대리인 : 장성구, 김원준

심사청구 : 있음

(54) 오버랩된 움직임 보상장치 및 그 방법

요약

본 발명은 블럭정합 알고리즘을 이용한 영상 부호화시에 야기되는 블럭킹 현상을 방지할 수 있는 오버랩된 움직임 보상장치에 관한 것으로, 이를 위하여, 본 발명은, 현재 프레임의 각 블럭의 경계가 서로 인접하는 블럭의 경계와 서로 겹치도록 분할한 다음 이전 프레임의 각 블럭에 의거하여 움직임 보상을 수행하고, 이 움직임 보상된 각 블럭에 대하여 서로 겹치는 위치에 의거하여 인접 블럭들과의 평균을 이용하여 그 경계의 화소값을 구한 다음 이를 선택적으로 출력하는 기술적수단을 채용함으로써, 움직임 보상을 위해 분할되는 블럭들사이에서 야기되는 블럭킹 현상을 방지할 수가 있고, 또한 각 블럭에서 움직임 보상에 따른 에지의 불연속성등과 같은 노이즈를 줄일 수 있는 것이다.

대표도

도2

명세서

[발명의 명칭]

오버랩된 움직임 보상장치 및 그 방법

[도면의 간단한 설명]

제 2 도는 본 발명의 바람직한 실시예에 따른 오버랩된 움직임 보상장치에 대한 블럭구성도, 제 3 도는 본 발명에 따라 분할되는 각 블럭들이 서로 겹치게 분할되는 형태의 일예를 보여주는 도면, 제 4 도는 본 발명에 따라 움직임 보상된 후의 프레임의 각 화소값들을 구하는 과정을 설명하기 위해 일예로서 도시한 도면.

본 내용은 요부공개 건이므로 전문내용을 수록하지 않았음

(57) 청구의 범위

청구항 1

소정의 크기로 분할되는 이전 프레임의 블럭과 이에 대응하는 현재 프레임의 블럭간의 차를 이용하여 영상의 움직임을 보상하는 움직임 보상장치에 있어서, 그 경계가 서로 겹치도록 소정의 크기로 분할되는 상기 현재 프레임의 각 블럭에 대한 움직임 벡터에 의거하여 해당 블럭의 어드레스를 각각 발생하기 위한 복수의 어드레스 발생수단; 상기 현재 프레임의 블럭에 각각 대응하는 이전 프레임의 블럭이 저장되며 상기 복수의 어드레스 발생수단에 대응하는 수의 메모리로 구성되어 상기 현재 프레임의 각 블럭에 대한 움직임 보상된 블럭을 출력하는 복수의 프레임 메모리수단; 상기 움직임 보상된 인접 블럭들간에 서로 겹치는 부분에 대한 화소값을 산출하기 위하여 상기 각 프레임 메모리수단의 출력에 가감산을 수행하여 서로 다른 복수의 출력신호를 발생하는 출력절환수단; 및 서로 겹치는 위치에 의거하여 상기 출력절환수단으로 부터의 상기 복수의 출력신호를 선택적으로 절환하여 움직임 보상된 프레임 데이터를 출력하는 멀티플렉서수단으로 이루어진 오버랩된 움직임 보상장치.

청구항 2

제 1 항에 있어서, 상기 복수의 어드레스 발생수단과 복수의 프레임 메모리수단은 각각 대응하는 4 개의 어드레스 발생기와 4 개의 프레임 메모리로 구성된 것을 특징으로 하는 오버랩된 움직임 보상장치.

청구항 3

제 2 항에 있어서, 상기 출력절환수단은, 상기 제1 및 제2프레임 메모리의 출력을 가산하는 제1가산기; 상기 제1가산기의 출력을 1/2 감산하여 상기 멀티플렉서에 상기 복수의 출력신호중의 하나로서 제공하는 제1감산기; 상기 제1 및 제3프레임 메모리의 출력을 가산하는 제2가산기; 상기 제2가산기의 출력을 1/2 감산하여 상기 멀티플렉서에 상기 복수의 출력신호중의 다른 하나로서 제공하는 제2감산기; 상기 제1, 제2, 제3 및 제4프레임 메모리의 각 출력을 가산하는 제3가산기; 및 상기 제3가산기의 출력을 1/4 감산하여 상기 멀티플렉서에 상기 복수의 출력신호중의 또다른 하나로서 제공하는 제3감산기로 구성된 것을 특징으로 하는 오버랩된 움직임 보상장치.

청구항 4

소정의 크기로 분할되는 이전 프레임의 블록과 이에 대응하는 현재 프레임의 블록간의 차를 이용하여 영상의 움직임을 보상하는 움직임 보상방법에 있어서, 상기 현재 프레임의 각 블록의 경계가 서로 인접하는 블록의 경계와 서로 겹치도록 분할한 다음 상기 이전 프레임의 각 블록에 의거하여 움직임을 보상할 수 행하고, 이 움직임 보상된 각 블록에 대하여 서로 겹치는 위치에 의거하여 인접 블록들과의 평균을 이용하여 그 경계의 화소값을 구한 다음 이를 선택적으로 출력하는 것을 특징으로 하는 오버랩된 움직임 보상장치.

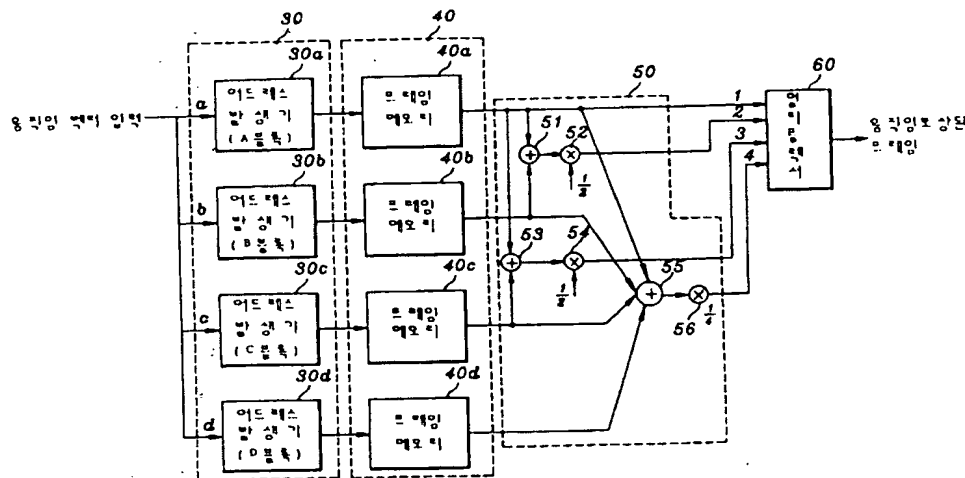
청구항 5

제 4 항에 있어서, '상기 각 블록의 경계에 대한 화소값들은 소정의 크기로 분할되는 각 블록의 움직임 보상된 출력값의 가감산 연산을 통해 산출하는 것을 특징으로 하는 오버랩된 움직임 보상장치.

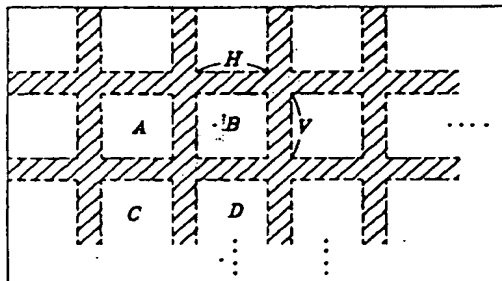
※ 참고사항 : 최초출원 내용에 의하여 공개하는 것임.

도면

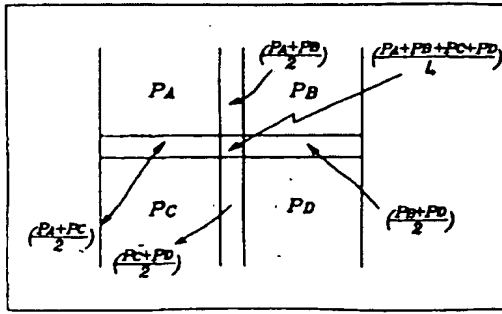
도면2



도면3



도면4





KOREAN PATENT ABSTRACTS(KR)

Document Code:A

(11) Publication No.1020010045433

(43) Publication.Date. 20010605

(21) Application No.1019990048722

(22) Application Date. 19991105

(51) IPC Code:

H04N 7/247

(71) Applicant:

DAEWOO ELECTRONICS CO., LTD.

(72) Inventor:

JANG, DONG IL

(30) Priority:

(54) Title of Invention

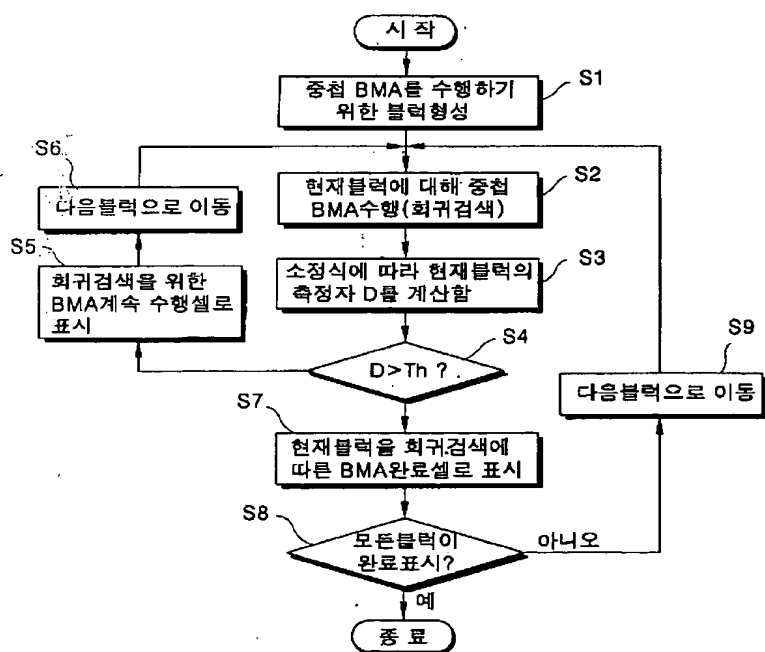
METHOD FOR ESTIMATING MOTION USING OVERLAPPED BLOCK MATCHING ALGORITHM

Representative drawing

(57) Abstract:

PURPOSE: A method for estimating motion using an overlapped block matching algorithm is provided to enhance a processing velocity thereof by reducing the number of blocks used in a circularly and repeatedly revolution search.

CONSTITUTION: A block is formed by separating the inputted image frame into a predetermined block for performing an overlapped block matching algorithm(S1). A revolution search is performed in accordance with the overlapped block matching algorithm(BMA) with respect to the current block(S2). A measuring unit of the current block is calculated(S3). The current calculating value is compared with the reference value(Th ,S4). If the calculating value is more than the



reference value(Th), the current block is displayed as BMA continuous cell for a revolution search of the current block($S5$). The process is repeated with respect to the next block($S6$) from the above ($S2$) stage. If the calculating value is less than the reference value(Th), the current block is displayed as BMA complete cell in accordance with a revolution search of the current block($S7$). It is judged whether all blocks are displayed as a complete cell($S8$). If all blocks are displayed as a complete cell, the revolution search is complected. If all blocks are not displayed as a complete cell, the process is moved to the next block and repeats the revolution search(S).

COPYRIGHT 2001 KIPO

if display of image is failed, press (F5)

(19) 대한민국특허청(KR) (12) 공개특허공보(A)

(51) Int. Cl.⁶ (11) 공개번호 특2001-0045433
H04N 7/247 (43) 공개일자 2001년06월05일

(21) 출원번호 10-1999-0048722
(22) 출원일자 1999년11월05일
(71) 출원인 대우전자 주식회사
서울 중구 남대문로5가 541
(72) 발명자 장동일
서울특별시용산구이촌1동강촌아파트106동1401호
(74) 대리인 진천웅, 윤창일

심사청구 : 있음

(54) 중첩 블럭정합알고리즘을 이용한 움직임 추정방법

요약

본 발명은 중첩 블럭정합알고리즘(overlapped BMA)을 이용한 움직임 추정방법에 관한 것이다.

이러한 본 발명의 방법은 중첩 블럭정합알고리즘을 수행하기 위해 입력 영상프레임을 소정의 블럭으로 구분하여 블럭을 형성하는 단계와, 현재블럭에 대해 중첩 블럭정합알고리즘(BMA)에 따라 회귀검색을 수행하는 단계, 소정 판별식에 따라 현재블럭의 측정자(D)를 연산하고, 연산결과값이 기준치보다 크면 현재블럭을 회귀검색을 위한 BMA 계속셀로 표시하는 단계, 연산결과값이 기준치보다 작으면 현재블럭을 회귀검색에 따른 BMA완료셀로 표시하고, 모든 블럭이 완료셀로 표시되면 회귀검색을 종료하는 단계, 및 상기 BMA계속 셀 혹은 BMA완료 셀 표시후 다음 블럭에 대해 상기 회귀검색을 반복하는 단계를 포함한다.

따라서, 본 발명에 따르면 중첩 블럭정합알고리즘에 따라 움직임 벡터를 구할 경우에 현재 회귀검색결과가 자신과 주변블럭에 미친 영향을 고려한 후 그 영향의 정도가 소정치 이하일 경우에는 해당 셀을 회귀검색에서 제외하여 점차 회귀검색에 사용되는 블럭의 수를 줄임으로써 처리속도를 향상시킬 수 있다.

대표도

도2

색인어

중첩블럭정합알고리즘 회귀검색

명세서

도면의 간단한 설명

도 1은 블럭정합알고리즘(BMA)을 설명하기 위하여 도시한 개념도,

도 2는 본 발명에 따른 움직임 추정방법의 절차를 도시한 도면이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 중첩 블럭정합알고리즘(overlapped BMA)을 이용한 움직임 추정방법에 관한 것으로, 특히 중첩 블럭정합알고리즘을 수행함에 있어 현재 회귀검색이 자신 및 인접블럭에 미친영향을 고려하여 회귀검색에 참여되는 블럭의 수를 점차줄여 처리속도를 개선한 움직임 추정방법에 관한 것이다.

일반적으로, 영상신호 처리기술에서 '움직임 추정(motion estimation)'이란 연속되는 영상신호에서 현재 프레임(current frame)의 화소(pixel)들이 이전 프레임(previous frame)에 비해 어느 정도 움직였는지를 벡터로 표시한 움직임 벡터 (motion vector)를 추정하여, 전체 영상을 전송하는 대신에, 이들 움직임 벡터를 전송함으로써 전송정보를 줄이는 기술(즉, 영상압축)을 말하는 것이다.

이러한 움직임 추정에서 널리 사용되는 블럭 정합 알고리즘(BMA: Block Matching Algorithm)은 도 1에 도시된 바와 같이, 화면의 움직임이 수평 또는 수직으로 평행 이동한 것으로 가정하여, 움직임이 일어난 프레임(즉, 현재 프레임)의 블럭영상이 움직임이 일어나기전 프레임(즉, 이전 프레임)의 어느 위치에 있는 블럭영상과 가장 일치하는가를 추정하여 그 위치를 움직임 벡터로서 추정하는 방법이다. 이때, 블럭의 크기로는 8 x 8, 16 x 16(가로 픽셀 수 x 세로 픽셀 수)을 주로 사용한다. 여기서, 현재 프레임

의 레퍼런스 블록(reference block 혹은 current block)과 가장 유사한 이전 블록을 찾기 위하여 이전 프레임(previous frame)에서 레퍼런스 블록의 위치를 중심으로 일정 범위 안을 찾게 되는데, 이러한 범위를 "서치 윈도우(search window 혹은 area)"라 하고, 이러한 서치 윈도우안에서 각 후보블록(candidate block)과의 차를 디스토션(distortion; 또는 절대에러(AE)라고도 한다)이라 하며, 두 블록간의 유사정도를 나타낸다.

또한, 서치 윈도우안의 모든 후보블록과 레퍼런스 블록을 비교하는 것을 "풀 서치 블록매칭(full search block matching algorithm)"이라 하며, 수식적으로 가장 일치하는 블록을 찾기 위하여 연속된 두 프레임의 영상중에서, 이전 프레임을 $f_1(x,y)$, 현재 프레임을 $f_2(x,y)$ 라 했을 때, $f_2(x,y)$ 와 $f_1(x-a, y-b)$ 에서 a,b를 변화시켜 가면서 $f_1(x-a,y-b)$ 와 $f_2(x,y)$ 의 차를 구하여, 그 차가 최소가 되는 (a,b)를 움직임 벡터로 예측한다. 이와 같이 블록간의 최소오차를 구하는 방법을 평균절대오차(MAE:Mean Absolute Error) 방법이라 한다.

그런데 이러한 블록매칭알고리즘(BMA)은 널리 알려진 바와 같이, 블록단위로 획일적으로 움직임 벡터를 구하기 때문에 구획효과(Blocking effect)가 발생되어 화질이 열화되는 문제점이 있다. 이러한 문제점을 해소하기 위하여 중첩 블록매칭알고리즘은 각 블록의 경계면을 중첩되게 하여 구획효과를 억제한다.

그러나 중첩 블록매칭알고리즘에서는 각 블록의 픽셀들의 움직임벡터가 인접한 픽셀들의 움직임 벡터 값에 영향을 받는다. 따라서 중첩 블록매칭알고리즘에 따라 움직임 벡터를 구하기 위해서는 각 픽셀들에 초기값을 부여한 후, 프레임의 시작부분부터 끝부분까지 순차적으로 최적화 과정을 진행한다. 이때 최적화된 픽셀들의 움직임 벡터값도 다음 픽셀의 움직임 벡터값이 최적화되면서 변환에 따라 다시 최적화될 필요가 있기 때문에, 각 픽셀들에 대한 최적화가 진행되면서 이전 픽셀들의 움직임 벡터값도 연쇄적으로 변하게 된다. 즉, 종래에는 픽셀들의 움직임 벡터를 최적화하는 과정에서 인접하는 픽셀들의 값이 변환에 따라 한 프레임의 전체 픽셀들에 대한 최적화가 완료될 때까지 순환반복적인 검색이 필요하게 되어 계산량이 증가하게 되는 문제점이 있다.

발명이 이루고자하는 기술적 과제

본 발명은 상기와 같은 문제점을 해소하기 위하여 중첩 블록매칭알고리즘에서 소정의 판별식에 따라 순환반복적인 회귀검색에 사용되는 블록의 수를 점차 줄여감으로써 처리속도를 향상시킨 중첩 블록매칭알고리즘을 이용한 움직임 추정방법을 제공하는데 그 목적이 있다.

상기와 같은 목적을 달성하기 위하여 본 발명의 방법은, 중첩 블록매칭알고리즘을 수행하기 위해 입력 영상프레임을 소정의 블록으로 구분하여 블록을 형성하는 단계; 현재블록에 대해 중첩 블록매칭알고리즘(BMA)에 따라 회귀검색을 수행하는 단계; 소정 판별식에 따라 현재블록의 측정자(D)를 연산하고, 연산결과값이 기준치보다 크면 현재블록을 회귀검색을 위한 BMA 계속셀로 표시하는 단계; 연산결과값이 기준치보다 작으면 현재블록을 회귀검색에 따른 BMA완료셀로 표시하고, 모든 블록이 완료셀로 표시되면 회귀검색을 종료하는 단계; 및 상기 BMA계속 셀 혹은 BMA완료 셀 표시후 다음 블록에 대해 상기 회귀검색을 반복하는 단계를 포함하는 것을 특징으로 한다.

발명의 구성 및 작용

이하에서는 첨부된 도면을 참조하여 본 발명의 실시예를 자세히 설명하기로 한다.

중첩 블록매칭알고리즘(overlapped BMA)은 한 블록을 둘러싼 네개의 블록들의 움직임 벡터 및 해당 블록의 움직임 벡터를 이용하여 각 블록의 예측값을 구성한다. 따라서 각 블록에서의 예측오차는 해당 블록의 움직임 오차는 물론 주위의 4개 움직임 벡터값에 의해서도 좌우된다. 이러한 다섯개의 블록간에는 상호 상관성이 존재하므로, 각 블록에서의 움직임 벡터값이 주변 4블록에 대한 예측오차에 영향을 미치게 된다.

그리고 각 블록 및 주변 4블록에서 움직임 벡터 집합은 회귀검색을 수행함에 따라 최적의 값에 근접하게 되는데, 각 블록에 할당된 움직임 벡터를 회귀검색을 통해 검색하므로써 얻게 되는 성능개선의 정도가 회귀검색 수행여부를 결정하는 가장 중요한 지수가 된다. 이와 같이 성능개선의 가능성을 판단하는 본 발명에 따른 측정자 D는 다음과 같이 정의된다.

$$D = D_1 + D_2$$

$$D_1 = \frac{1}{4} \left[\sum_{\substack{m,n=1 \\ m,n \neq 0}}^4 \{SAD(B_{i-m,j-n}^k) - SAD'(B_{i-m,j-n}^k)\} / SAD(B_{i-m,j-n}^k) \right]$$

$$D_2 = |SAD(B_{i,j}^{k+1}) - SAD(B_{i,j}^k)|^{1/\alpha}$$

상기 수학식1에서 $SAD(B^k)$ 는 블록 B에서 k번째 회귀검색시 얻어진 움직임 예측 오차를 의미한다. 이 값에서는 현재 블록 B의 움직임벡터에 의한 예측값도 포함된다. 반면, $SAD'(B^k)$ 는 해당 블록 B의 움직임 벡터에 의한 예측값을 사용하지 않은 경우의 오차를 의미한다. 따라서 수학식1의 D_1 항은 현재블록의 움직임벡터가 주위블록에 미치는 공헌도를 나타내고, 수학식1의 D_2 항은 금번 회귀검색을 통해서 해당 블록의 오차가 얼마나 변했는지를 나타낸다. 따라서 D_2 항의 값이 작을 수록 회귀검색의 효과가 없음을 나타낸다. 수학식1에서 양수 α 는 D_1 항과 D_2 항의 영향을 조절하기 위한 상수로 응용에 따라 적절히 결정한다.

이와 같이 수식식1에 따른 연산결과 0값은, 주위블럭과의 상관성과 자신의 회귀검색 결과를 동시에 고려할 수 있는 측정자이므로 이 측정자를 정렬에 이용하면 정확한 결과를 구할 수 있다.

도 2는 본 발명에 따라 움직임 추정을 수행하는 과정을 도시한 흐름도이다.

도 2를 참조하면, 단계 S1에서는 중첩 블럭정합알고리즘을 수행하기 위해 입력 영상프레임을 소정의 블럭으로 구분하여 블럭을 형성하고, 단계 S2에서는 현재블럭에 대해 중첩 블럭정합알고리즘(BMA)에 따라 회귀검색을 수행한다.

그리고 단계 S3에서는 앞서 설명한 바와 같이, 수식식1에 따라 현재블럭의 측정자(D)를 연산하고, 단계 S4에서 연산결과값을 기준치(Th)와 비교한다. 비교결과 연산값(D)이 기준치(Th)보다 크면, 단계 S5에서 현재블럭을 회귀검색을 위한 "BMA 계속 셀"로 표시하고, 단계 S6에서 다음 블럭으로 이동한 후 다음 블럭에 대해서 단계 S2부터 반복한다.

한편, 단계 S4에서 연산값(D)이 기준치(Th)보다 작으면 현재블럭을 회귀검색에 따른 "BMA완료 셀"로 표시하고, 단계 S8에서는 모든 블럭이 완료셀로 표시되었는지를 판단한 후 완료셀로 표시되면 회귀검색을 종료하고, 아니면 단계 S9에서 다음 블럭으로 이동한 후 상기 회귀검색을 반복한다.

발명의 효과

이상에서 설명한 바와 같이, 본 발명에 따르면 중첩 블럭정합알고리즘에 따라 움직임 벡터를 구할 경우에 현재 회귀검색 결과가 자신과 주변블럭에 미친 영향을 고려한 후 그 영향의 정도가 소정치 이하일 경우에는 해당 셀을 회귀검색에서 제외하여 회귀검색에 사용되는 블럭의 수를 점차 줄임으로써 처리속도를 향상시킬 수 있는 효과가 있다.

(57) 청구의 범위

청구항 1

중첩 블럭정합알고리즘을 수행하기 위해 입력 영상프레임을 소정의 블럭으로 구분하여 블럭을 형성하는 단계;

현재블럭에 대해 중첩 블럭정합알고리즘(BMA)에 따라 회귀검색을 수행하는 단계;

소정 판별식에 따라 현재블럭의 측정자(D)를 연산하고, 연산결과값이 기준치보다 크면 현재블럭을 회귀검색을 위한 BMA 계속셀로 표시하는 단계;

연산결과값이 기준치보다 작으면 현재블럭을 회귀검색에 따른 BMA완료셀로 표시하고, 모든 블럭이 완료셀로 표시되면 회귀검색을 종료하는 단계; 및

상기 BAM계속 셀 혹은 BMA완료 셀 표시후 다음 블럭에 대해 상기 회귀검색을 반복하는 단계를 포함하는 중첩 블럭정합알고리즘을 이용한 움직임 추정방법.

청구항 2

$$D = D_1 + D_2$$

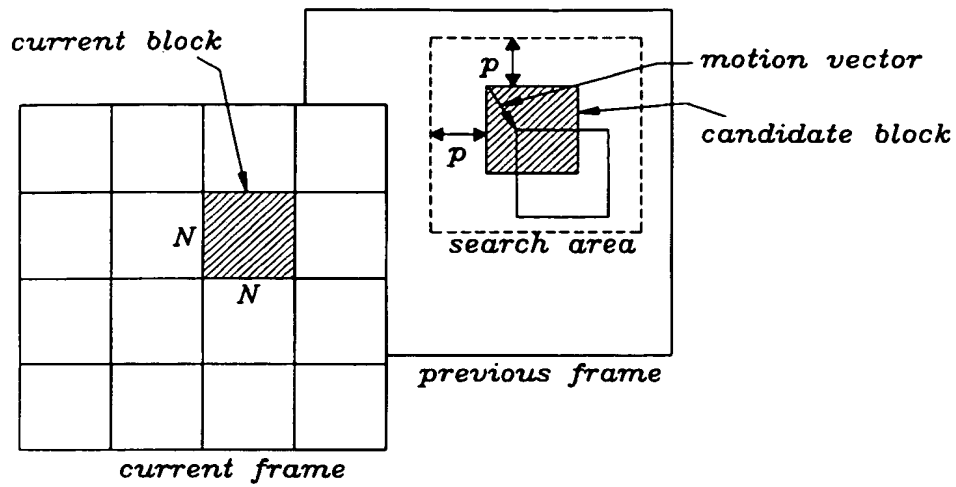
$$D_1 = \frac{1}{4} \cdot \left[\sum_{\substack{m,n=-1 \\ m,n=0}}^1 \{SAD(B_{i-m,j-n}^k) - SAD^*(B_{i-m,j-n}^k)\} \right] / SAD(B_{i-m,j-n}^k)$$

$$D_2 = |SAD(B_{i,j}^{k-1}) - SAD(B_{i,j}^k)|^{\alpha}$$

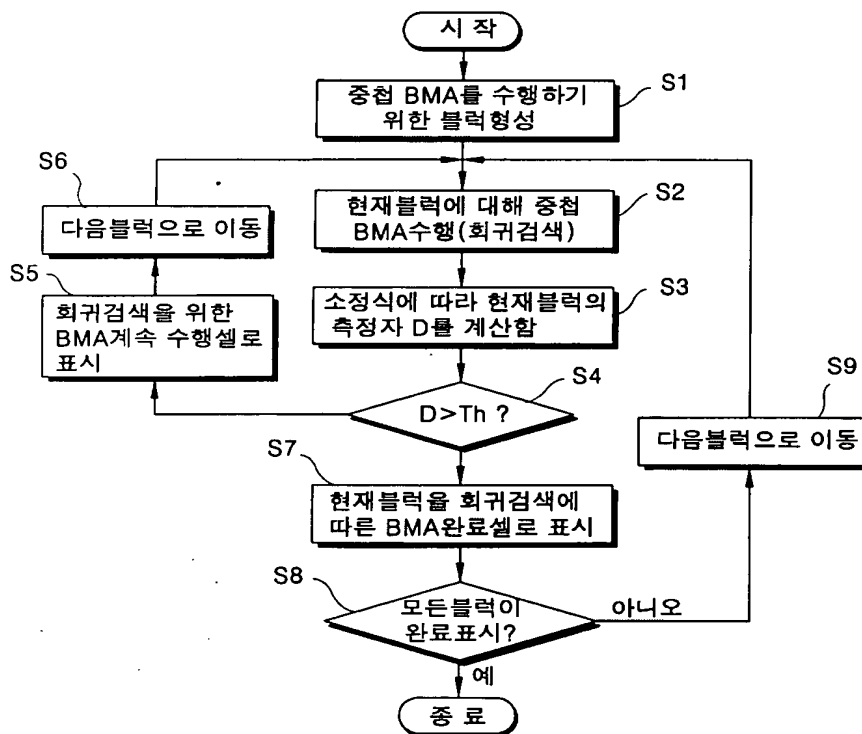
제1항에 있어서, 상기 판별식은 인 것을 특징으로
하는 중첩 블럭정합알고리즘을 이용한 움직임 추정방법.

도면

도면1



도면2



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-299864

(43)Date of publication of application : 24.10.2000

(51)Int.Cl.

H04N 7/32

(21)Application number : 11-104088

(71)Applicant : CANON INC

(22)Date of filing : 12.04.1999

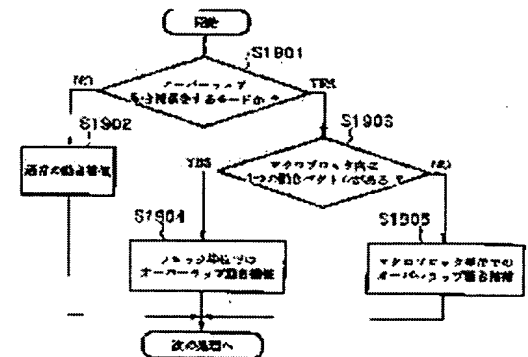
(72)Inventor : NAKAYAMA TADAYOSHI

(54) METHOD FOR PROCESSING DYNAMIC IMAGE

(57)Abstract:

PROBLEM TO BE SOLVED: To fast perform overlap motion compensation that reduces discontinuous distortions between adjacent blocks caused by motion vector prediction by generating a weight coefficient matrix corresponding to a macroblock and performing overlap motion compensation prediction coding in a macroblock unit by using it.

SOLUTION: It is identified whether or not a mode for performing overlap motion compensation is (S1901), and when it is not the mode, normal motion compensation is executed (S1902). Meanwhile, in the case four motion vectors exist in a macroblock when it is decided as a mode performing overlap motion compensation, overlap motion compensation processing in a general block unit is performed (S1904). In the case not four motion blocks but only one block exists, motion compensation processing is performed in a macroblock unit (S1905). Thus, it is possible to reduce the number of pixels to be accessed and the number of add operations.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-299864

(P2000-299864A)

(43) 公開日 平成12年10月24日 (2000.10.24)

(51) Int. Cl.

H04N 7/32

識別記号

F I

H04N 7/137

データベース(参考)

Z 5C059

審査請求 未請求 請求項の数20 O L (全 13 頁)

(21) 出願番号 特願平11-104088

(22) 出願日 平成11年4月12日 (1999.4.12)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 中山 忠義

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(74) 代理人 100076428

弁理士 大塚 康徳 (外2名)

Fターム(参考) 5C059 KK03 KK15 M000 MA04 MA23

NN01 NN28 NN30 PP04 SS06

SS20 SS26 TA61 TB07 TC12

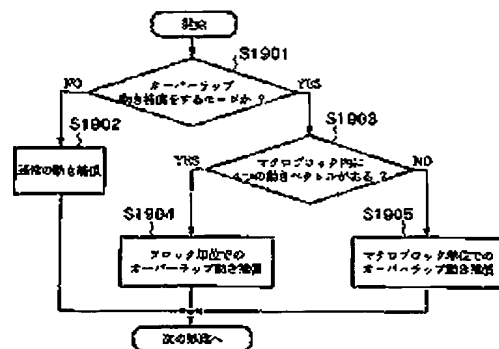
TC42 TC43 TD05 UA02 UA33

(54) 【発明の名称】 動画処理方法

(57) 【要約】

【課題】 オーバーラップ動き補償予測符号化処理は、加算演算の回数が多いために処理時間がかかってしまう。

【解決手段】 マクロブロック内の動きベクトルが同一であれば、ステップS1905においてマクロブロック単位でのオーバーラップ動き補償を行なうが、その際に、複数の重み係数マトリクスにおいて有為係数のみの演算を行なえば良いため、実質的な演算量を削減することができ、処理速度が向上する。



【特許請求の範囲】

【請求項1】 着目するブロック及びそれに隣接するブロックにおける動きベクトルのそれぞれに基づく第1の重み係数マトリクスより、複数ブロックからなるマクロブロックに対応する第2の重み係数マトリクスを生成し、

該第2の重み係数マトリクスを用いてマクロブロック単位でオーバーラップ動き補償予測符号化を行なうことを特徴とする動画画像処理方法。

【請求項2】 前記オーバーラップ動き補償予測符号化の際に、前記第2の重み係数マトリクスに基づいて前記マクロブロックの予測画像を導出することを特徴とする請求項1記載の動画画像処理方法。

【請求項3】 前記第1の重み係数マトリクスは前記ブロックのサイズに対応し、前記第2の重み係数マトリクスは前記マクロブロックのサイズに対応することを特徴とする請求項1記載の動画画像処理方法。

【請求項4】 前記マクロブロックは、4つの前記ブロックよりなることを特徴とする請求項3記載の動画画像処理方法。

【請求項5】 前記第2の重み係数マトリクスは、複数の前記第1の重み係数マトリクスを加算し、前記マクロブロックのサイズに連結することによって生成されることを特徴とする請求項3記載の動画画像処理方法。

【請求項6】 前記隣接するブロックは、前記着目するブロックの上下左右に隣接するブロックであることを特徴とする請求項1記載の動画画像処理方法。

【請求項7】 前記第2の重み係数マトリクスは、着目するマクロブロック及びその上、左、右に隣接するマクロブロックにおける動きベクトルのそれぞれに対応して複数生成されることを特徴とする請求項1記載の動画画像処理方法。

【請求項8】 前記オーバーラップ動き補償予測符号化の際に、前記第2の重み係数マトリクスの有為係数に關してのみ重み付け加算演算が行われることを特徴とする請求項1記載の動画画像処理方法。

【請求項9】 前記有為係数は、0又は8以外の係数であることを特徴とする請求項8記載の動画画像処理方法。

【請求項10】 前記第2の重み係数マトリクスは、前記有為係数のみが保持されることを特徴とする請求項8記載の動画画像処理方法。

【請求項11】 前記着目するマクロブロックは、1つの動きベクトルを有することを特徴とする請求項1記載の動画画像処理方法。

【請求項12】 前記オーバーラップ動き補償予測符号化の際に、前記着目するマクロブロックの所定領域毎に、前記第2の重み係数マトリクスに基づく演算を施すことを特徴とする請求項11記載の動画画像処理方法。

【請求項13】 前記隣接するマクロブロックの全てがそれぞれ1つの動きベクトルを有するか否かに応じて、

前記所定領域が異なることを特徴とする請求項12記載の動画画像処理方法。

【請求項14】 前記隣接するマクロブロックの少なくとも1つが複数の動きベクトルを有する場合、前記所定領域は前記ブロックのサイズに対応することを特徴とする請求項13記載の動画画像処理方法。

【請求項15】 更に、前記着目するマクロブロックが2つ以上の動きベクトルを有するか否かを判定し、

該判定結果が否である場合に、前記オーバーラップ動き補償予測符号化を行なうことを特徴とする請求項11記載の動画画像処理方法。

【請求項16】 前記着目するマクロブロックが2つ以上の動きベクトルを有すると判定された場合に、前記第1の重み係数マトリクスを用いて前記ブロック単位によるオーバーラップ動き補償予測符号化を行なうことを特徴とする請求項15記載の動画画像処理方法。

【請求項17】 前記請求項1乃至16のいずれかに記載の動画画像処理方法により符号化された動画画像を復号する動画画像処理方法。

【請求項18】 前記請求項1乃至16のいずれかに記載の動画画像処理方法により動画画像を符号化する動画画像処理装置。

【請求項19】 前記請求項1乃至16のいずれかに記載の動画画像処理方法により符号化された動画画像を復号する動画画像処理装置。

【請求項20】 オーバーラップ動き補償予測符号化を行なう動画画像処理方法のプログラムコードが記録された記録媒体であって、該プログラムは、着目するブロック及びそれに隣接するブロックにおける動きベクトルのそれぞれに基づく第1の重み係数マトリクスより、複数ブロックからなるマクロブロックに対応する第2の重み係数マトリクスを生成するコードと、該第2の重み係数マトリクスを用いてマクロブロック単位でオーバーラップ動き補償予測符号化を行なうコードと、を含むことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は動画画像の動き補償予測符号化を行なう動画画像処理方法に関する。

【0002】

【従来の技術】近年、画像信号の符号化技術や該符号化データを通信したり蓄積したりする技術、また、該符号化データを安価なコストで復号するための半導体技術、復号した画像を表示する技術等、いわゆる画像情報を扱うインフラ技術の発達がめざましい。

【0003】これにより、動画画像のような大量の情報を有する画像であっても遠隔地へ転送することができ、多地点間におけるより高度な情報交換が可能になってきた。

【0004】動画画像の符号化においては、フレーム間の

画像信号の相関性を利用して情報量を大幅に削減することが可能である。この特性を利用した代表的な符号化技術として、所謂動き補償予測符号化方式が知られている。該符号化方式においては、所定の画素ブロック毎に、参照すべき画像の相対位置を表わす動きベクトルと称する情報を参照する。そして、該動きベクトル情報に基づいて得られる画素情報を予測値とし、該予測値と実際の画素値との差分に対してDCT等の直交変換を施すことにより、該変換情報を人間の視覚特性に基づいて削減することができる。

【0005】このようなブロック単位での符号化方式において、圧縮率を上げるために変換情報を大幅に削減すると、ブロック状の歪が出現しやすくなってしまふ。一般に符号の転送レートが低い装置において使用する圧縮方式においては、圧縮率を上げることは必須であるため、ブロック状の歪は避け難いものとなっていた。

【0006】そこで、ITU-T勧告H.263規格やMPEG4規格においては、前記ブロック状の歪を軽減するため、オーバーラップ動き補償という処理を行なっている。

【0007】一般的な動き補償予測においては、フレーム画像内の着目ブロックを、参照画像上で動きベクトル分だけずらしたものを該着目ブロックの予測画像とすることにより、符号化を行なう。この場合、隣接ブロック間で動きベクトルが同一であれば、該隣接する2つのブロックの予測画像データは隣接部において連続性があるため、予測画像データ上ではブロック歪が生じることは無い。ところが、隣接ブロック間で動きベクトルに若干の差がある場合、その動きベクトルの差が、隣接する2つのブロックの予測画像の隣接部における不連続性となって現れる。

【0008】そこで、着目ブロックの動きベクトルのみを用いて予測を行なう代わりに、隣接するブロックの動きベクトルを反映するような予測を行なうことによって、上記不連続性を軽減することが考えられる。

【0009】オーバーラップ動き補償では、着目ブロックの動きベクトルを用いて予測した画像データ(1ブロック分)、隣接する4つのブロックの動きベクトルを用いて予測した画像データ(1ブロック分×4)の各々に重み係数を乗じ、それらを台算したデータを最終的な予測値(画像)とする。

【0010】

【発明が解決しようとする課題】しかしながら、従来のオーバーラップ動き補償においては、加算演算の回数が多く、処理時間がかかってしまうといった問題があった。

【0011】本発明は上記問題を解決するためになされたものであり、動きベクトル予測に伴って生じる隣接ブロック間の不連続性の歪を軽減するオーバーラップ動き補償を高速に可能とする動画処理方法を提供することを目的とする。

【0012】

【課題を解決するための手段】上記目的を達成するための一手法として、本発明の動画処理方法は以下の工程を備える。

【0013】即ち、着目するブロック及びそれに隣接するブロックにおける動きベクトルのそれぞれに基づく第1の重み係数マトリクスより、複数ブロックからなるマクロブロックに対応する第2の重み係数マトリクスを生成し、該第2の重み係数マトリクスを用いてマクロブロック単位でオーバーラップ動き補償予測符号化を行なうことを特徴とする。

【0014】

【発明の実施の形態】以下、本発明に係る一実施形態について詳細に説明する。

【0015】＜第1実施形態＞まず、一般的なオーバーラップ動き補償について簡単に説明する。

【0016】図1において、符号化または復号対象であるフレーム画像101内の座標(XG, YG)を位置情報とする着目ブロック102を、参照画像104上で動きベクトル103分だけずらした位置にある1ブロックの画像データを、着目ブロック102の予測画像105とする。これが、一般的な動き補償による予測である。

【0017】このような動き補償予測において、隣接ブロック間で動きベクトルが同一であれば、該隣接する2つのブロックの予測画像データは隣接部において連続性があるため、予測画像データ上ではブロック歪が生じることは無い。ところが、隣接ブロック間で動きベクトルに若干の差がある場合、その動きベクトルの差が、隣接する2つのブロックの予測画像の隣接部における不連続性となって現れる。

【0018】そこで、着目ブロックの動きベクトルのみを用いて予測を行なう代わりに、隣接するブロックの動きベクトルを反映するような予測を行なうことによって、上記不連続性を軽減することが考えられる。

【0019】オーバーラップ動き補償では、着目ブロックの動きベクトルを用いて予測した画像データ(1ブロック分)、隣接する4つのブロックの動きベクトルを用いて予測した画像データ(1ブロック分×4)の各々に重み係数を乗じ、それらを台算したデータを最終的な予測値(画像)とする。

【0020】具体的には、着目ブロックの動きベクトルを用いて予測した画像データに対しては図2に示す重み係数マトリクスを乗じ、また、着目ブロックの上、下、左、右のそれぞれに隣接するブロックの動きベクトルを用いて予測した画像データに対してはそれぞれ図3、図4、図5、図6に示す重み係数マトリクスを乗じる。そして、それら4画像分を全て足し合わせた後に更に4を加算し、8で割った値を予測値(画像)とする。

【0021】上記演算は即ち、以下に示す(1)式で表される。

【0022】

$$p(x,y)=(q(x,y)H2(i,i)+u(x,y)H3(i,i)+d(x,y)H4(i,i)+l(x,y)H5(i,i)+r(x,y)H6(i,i)+4)/8 \quad \dots (1)$$

ここで、 (x,y) は画像全体における着目ブロック中の各画素位置を表わす座標であり、 i, j は着目ブロックの左上の座標を $(x0,y0)$ とした時、 $i=x-x0$ 、 $j=y-y0$ という値をとる。また、 $q(x,y)$ は着目ブロックの動きベクトルを用いて予測した画像データであり、 $u(x,y)$ 、 $d(x,y)$ 、 $l(x,y)$ 、 $r(x,y)$ はそれぞれ、着目ブロックの上、下、左、右に隣接するブロックの動きベクトルを用いて予測した画像データである。これらの画像データ $q(x,y)$ 、 $u(x,y)$ 、 $d(x,y)$ 、 $l(x,y)$ 、 $r(x,y)$ は、それぞれの動きベクトルを $(M1x,M1y)$ 、 $(M2x,M2y)$ 、 $(M3x,M3y)$ 、 $(M4x,M4y)$ 、 $(M5x,M5y)$ とし、予測の基になる参照画像を $ref(i,i)$ とすると、以下の様に表される。

【0023】 $q(x,y)=ref(x+M1x,y+M1y)$ $u(x,y)=ref(x+M2x,y+M2y)$ $d(x,y)=ref(x+M3x,y+M3y)$ $l(x,y)=ref(x+M4x,y+M4y)$ $r(x,y)=ref(x+M5x,y+M5y)$

また、 $H2(i,i)$ 、 $H3(i,i)$ 、 $H4(i,i)$ 、 $H5(i,i)$ 、 $H6(i,i)$ は、各々図2、図3、図4、図5、図6に示す重み係数マトリクスであり、それぞれ $0 \leq i, j \leq 7$ の範囲においてのみ、値が定義されている。

【0024】以下、本実施形態におけるオーバーラップ動き補償について説明する。本実施形態においては、マクロブロック単位によるオーバーラップ動き補償を行なうことを特徴とする。

【0025】ITU-T勧告H.263規格やMPEG4規格において *

$$p(x,y)=(q(x,y)H2(i,i)+u(x,y)H3(i,i)+q(x,y)H4(i,i)+l(x,y)H5(i,i)+q(x,y)H6(i,i)+4)/8 \\ = (q(x,y)(H2(i,i)+H4(i,i)+H6(i,i))+u(x,y)H3(i,i)+l(x,y)H5(i,i)+4)/8 \quad \dots (2)$$

同様にブロックbについては、(1)式において $d(x,y)$ と $l(x,y)$ を $q(x,y)$ に置き換えることができるため、以下の *

$$p(x,y)=(q(x,y)H2(i,i)+u(x,y)H3(i,i)+q(x,y)H4(i,i)+q(x,y)H5(i,i)+r(x,y)H6(i,i)+4)/8 \\ = (q(x,y)(H2(i,i)+H4(i,i)+H5(i,i))+u(x,y)H3(i,i)+r(x,y)H6(i,i)+4)/8 \quad \dots (3)$$

またブロックcについては、上に隣接するブロックaと右に隣接するブロックdが同一マクロブロック内にあるため、ブロックcと同じ動きベクトルを持つ。さらに下に隣接するブロックは符号化もしくは復号化処理がまだ行われておらず、動きベクトル情報が得られないため、ブ

$$p(x,y)=(q(x,y)H2(i,i)+q(x,y)H3(i,i)+q(x,y)H4(i,i)+l(x,y)H5(i,i)+q(x,y)H6(i,i)+4)/8 \\ = (q(x,y)(H2(i,i)+H3(i,i)+H4(i,i)+H6(i,i))+l(x,y)H5(i,i)+4)/8 \quad \dots (4)$$

同様にブロックdについては、 $u(x,y)$ 、 $d(x,y)$ 、 $l(x,y)$ を $q(x,y)$ に置き換えることができるため、以下の(5)式

*は、 8×8 のブロックという単位の上位に 15×16 のマクロブロックという単位があり、該マクロブロックは図7に示すように4つのブロックより成る。該マクロブロックにおいて、各ブロックがそれぞれ別の動きベクトルを持つことを可能とするモードが存在するが、マクロブロック全体における動きの相関が強い場合には、情報量削減のために該モードにおいても該マクロブロックを1つの動きベクトルで代表する場合がある。

【0026】以下、マクロブロック全体で1つの動きベクトルしか持たない、即ち、マクロブロックを構成する各ブロックの動きベクトルが等しい場合について考える。

【0027】上述した一般的なオーバーラップ動き補償予測を示す(1)式を、図7に示すマクロブロックを構成する4つのブロックa、b、c、dにそれぞれ適用すると、該(1)式は各ブロック毎に異なる演算式に変形される。

【0028】まず、ブロックaについて考える。ブロックaが着目ブロックであれば、ブロックbとブロックcはそれぞれ右と下に隣接するブロックであり、この場合マクロブロック内で1つの動きベクトルしか持たないため、ブロックbとブロックcはブロックaと同一の動きベクトルを持つことになる。従って、(1)式において $d(x,y)$ と $r(x,y)$ を $q(x,y)$ で置き換えることができる。よって、ブロックaに対するオーバーラップ動き補償予測については、以下の(2)式で示される演算式を得る。

【0029】

$$p(x,y)=(q(x,y)H2(i,i)+u(x,y)H3(i,i)+q(x,y)H4(i,i)+l(x,y)H5(i,i)+q(x,y)H6(i,i)+4)/8 \\ = (q(x,y)(H2(i,i)+H4(i,i)+H6(i,i))+u(x,y)H3(i,i)+l(x,y)H5(i,i)+4)/8 \quad \dots (2)$$

【0030】

$$p(x,y)=(q(x,y)H2(i,i)+u(x,y)H3(i,i)+q(x,y)H4(i,i)+q(x,y)H5(i,i)+r(x,y)H6(i,i)+4)/8 \\ = (q(x,y)(H2(i,i)+H4(i,i)+H5(i,i))+u(x,y)H3(i,i)+r(x,y)H6(i,i)+4)/8 \quad \dots (3)$$

ブロックcの動きベクトルで置き換えることができる。よって、前記(1)式において、 $u(x,y)$ 、 $d(x,y)$ 、 $r(x,y)$ を $q(x,y)$ に置き換えることができ、以下の(4)式で示される演算式を得る。

【0031】

$$p(x,y)=(q(x,y)H2(i,i)+q(x,y)H3(i,i)+q(x,y)H4(i,i)+l(x,y)H5(i,i)+q(x,y)H6(i,i)+4)/8 \\ = (q(x,y)(H2(i,i)+H3(i,i)+H4(i,i)+H6(i,i))+l(x,y)H5(i,i)+4)/8 \quad \dots (4)$$

で示される演算式を得る。

【0032】

$$\begin{aligned}
 p(x,y) &= (q(x,y)H2(i,j) + q(x,y)H3(i,j) + q(x,y)H4(i,j) \\
 &\quad + q(x,y)H5(i,j) + r(x,y)H6(i,j) + 4)/8 \\
 &= (q(x,y)(H2(i,j) + H3(i,j) + H4(i,j) + H5(i,j)) \\
 &\quad + r(x,y)H6(i,j) + 4)/8
 \end{aligned}$$

… (5)

以上の(2)～(5)式によれば、4つのブロックa、b、c、dにおける予測画像データ $q(x,y)$ に対する重み係数マトリクスは、それぞれ

$$H2(i,j) + H4(i,j) + H6(i,j)$$

$$H2(i,j) + H4(i,j) + H5(i,j)$$

$$H2(i,j) + H3(i,j) + H4(i,j) + H6(i,j)$$

$$H2(i,j) + H3(i,j) + H4(i,j) + H5(i,j)$$

となり、これらをそれぞれ図8(a)～(d)に示す。

【0033】該4つのブロックa、b、c、dをマクロブロック単位で処理しようとする、重み係数マトリクスもマクロブロックに対応した大きさ、すなわち 16×16 サイズとする必要がある。該重み係数マトリクスとしては、図8(a)～(d)に示した4つの重み係数マトリクスを図9に示すように連結することで得られる。ここで、このマトリクスを $M2(i,j)$ と定義する。

【0034】また、上に隣接するブロックの動きベクトルを用いて予測した画像データ $u(x,y)$ に対応する 16×16 の重み係数マトリクスは、図3に示す $H3(i,j)$ を2つ横に連結し、下半分である 16×8 の領域には0を埋め込んだも

$$\begin{aligned}
 p(x,y) &= (q(x,y)M2(i,j) + u(x,y)M3(i,j) + l(x,y)M5(i,j) \\
 &\quad + r(x,y)M6(i,j) + 4)/8
 \end{aligned}$$

… (6)

(6)式によれば、上記(1)式において0～7の範囲であった i, j の値が、0～15まで広がる。ここで(6)式によれば、図9に示す $M2(i,j)$ の3/8を占める。係数が8である要素については、 $p(x,y) = q(x,y)$ となるので、計算を行なう必要がなくなる。即ち、実際に演算処理を実行しなければならない要素が、全体の5/8まで減少することが分かる。

【0040】このように、本実施形態のマクロブロック単位によるオーバーラップ動き補償により、マクロブロックに対応した重み係数マトリクスにおいて有為係数のみの演算を行えば良いため、一般的なブロック単位によるオーバーラップ動き補償と比べて演算量を実質的に削減することができる。

【0041】以上の説明においては、着目マクロブロックは1つの動きベクトルしか持っていないという前提に基づき、新たなオーバーラップ動き補償の演算式や重み係数マトリクスを導出してきた。しかしながら実際には、上または左右に隣接する3つのマクロブロックも各々1つの動きベクトルしか持たない場合や、各々が4つの動きベクトルを持つ場合、あるいは、各々のブロックによって動きベクトル数が異なる場合などが考えられる。

【0042】従って本実施形態のオーバーラップ動き補償においては、上または左右に隣接する3つのマクロブロックの各々が、1つの動きベクトルしかない場合とそうでない場合とのそれぞれについて、異なる処理を行な

* ので表される。これを $M3(i,j)$ と定義し、図10に示す。

【0035】また、左に隣接するブロックの動きベクトルを用いて予測した画像データ $l(x,y)$ に対応する 16×16 の重み係数マトリクスは、図5に示す $H5(i,j)$ を2つ縦に連結し、右半分である 8×16 の領域には0を埋め込んだもので表される。これを $M5(i,j)$ と定義し、図11に示す。

【0036】また、右に隣接するブロックの動きベクトルを用いて予測した画像データ $r(x,y)$ に対応する 16×16 の重み係数マトリクスは、 $M5(i,j)$ を左右反転したもので表される。これを $M6(i,j)$ と定義し、図12に示す。

【0037】尚、下に隣接するブロックの動きベクトルを用いて予測した画像データ $d(x,y)$ に対応する重み係数マトリクスは、すでに $M2(i,j)$ に反映済みである。

【0038】これら重み係数マトリクス $M2(i,j)$ 、 $M3(i,j)$ 、 $M5(i,j)$ 、 $M6(i,j)$ を用いて、マクロブロック単位のオーバーラップ動き補償の演算式を記述すると、(6)式のようなになる。

【0039】

うことを特徴とする。

【0043】図13に、本実施形態における動き補償処理の概要フローチャートを示す。まずステップS1901において、オーバーラップ動き補償を行なうモードであるかを識別し、該モードでなければ、S1902において通常の動き補償を実行する。ここでは、重み付け演算は行なわず、

$$p(x,y) = q(x,y) = \text{ref}(x) + M1x + M1y$$

により予測画像を生成する。

【0044】一方、オーバーラップ動き補償を行なうモードであると識別された場合には、ステップS1903において、マクロブロック内に4つの動きベクトルがあるかを識別する。4つの動きベクトルがあれば、ステップS1904において(1)式に示した一般的なブロック単位によるオーバーラップ動き補償処理を行なう。一方、動きベクトルが4つではなく、1つしか無い場合には、ステップS1905において本実施形態の特徴であるマクロブロック単位でのオーバーラップ動き補償処理を行なう。

【0045】図14に、上記ステップS1905における本実施形態のマクロブロック単位によるオーバーラップ動き補償処理の詳細フローチャートを示す。

【0046】まずステップS2001において、上、左、右にそれぞれ隣接するマクロブロックにおける動きベクトルの数が、各々1つしか無いかを判別し、各々1つしか無い場合には、ステップS2002～S2005の処理を行な

う。

【0047】まずステップS2002においては、15×16の着目マクロブロックのうち、図15に示す領域21aについてのみ、

$p(x,y)=M3(1,j)u(x,y)+M3(1,i)ref(x+M/2x,y+M/2y)$
なる重み付け演算を行なう。次にステップS2003において、同じく図16に示す領域22aについてのみ、

$p(x,y)=p(x,y)+M5(1,j)l(x,y)=p(x,y)+M5(1,i)ref(x+M/4x,y+M/4y)$

なる重み付け演算を行なう。尚、該式においては右辺の $p(x,y)$ に対する加算処理により、左辺の $p(x,y)$ が新たに得られることを示す。そしてステップS2004において、図16に示す領域22bについてのみ、

$p(x,y)=p(x,y)+M6(1,j)l(x,y)=p(x,y)+M6(1,i)ref(x+M/5x,y+M/5y)$

なる重み付け演算を行なう。最後にステップS2005において、着目マクロブロックにおける図15に示す領域21a、図16に示す領域22a、22bに対して、

$p(x,y)=(p(x,y)+M2(1,i)q(x,y)+4)/8$
 $=(p(x,y)+M2(1,j)ref(x+M/1x,y+M/1y)+4)/8$

を演算し、着目マクロブロックの残りの領域に対しては、

$p(x,y)=q(x,y)=ref(x+M/1x,y+M/1y)$

のコピー処理を行なう。

【0048】上述したステップS2002～S2005の処理は、代入処理や加算処理を配慮すれば、その演算の順番を入れ替えることができ、例えばステップS2005の処理を最初に行なうことも可能である。この場合、上記ステップS2005における

$p(x,y)=(p(x,y)+M2(1,i)q(x,y)+4)/8$
 $=(p(x,y)+M2(1,j)ref(x+M/1x,y+M/1y)+4)/8$

なる加算処理は、

$p(x,y)=M2(1,j)q(x,y)$
 $+M2(1,i)ref(x+M/1x,y+M/1y)$

なる代入処理のみとなり、ステップS2002～S2004の他の処理ブロックで加算処理を行なうことになる。もちろん、演算の最後には、4を加算して8で割るといった丸め処理を行なう必要がある。

【0049】一方、ステップS2001における判別結果が異なる場合、即ち、隣接する3つのマクロブロックの内、少なくとも1つのマクロブロックが動きベクトルを4つ持っている場合、ステップS2010～S2016の処理を行なう。

【0050】この場合、隣接する3つのマクロブロックをブロックに分割し、図17に示すように、着目するマクロブロックとそれに隣接する6つのブロックU1、U2、L1、L2、R1、R2としてとらえる。

【0051】尚、同じマクロブロックに属する2つのブロックは、同一の動きベクトルを持つ（マクロブロック内に動きベクトルを1つしか持たない場合）こともあ

ば、異なる動きベクトルを持つ（マクロブロック内に4つの動きベクトルを持つ場合）こともある。ここで、6つのブロックU1、U2、L1、L2、R1、R2に対応する動きベクトルをそれぞれ、 $(M/b1x,M/b1y)$ 、 $(M/b2x,M/b2y)$ 、 $(M/b3x,M/b3y)$ 、 $(M/b4x,M/b4y)$ 、 $(M/b5x,M/b5y)$ 、 $(M/b6x,M/b6y)$ とする。

【0052】まずステップS2010においては、15×16の着目マクロブロックのうち、図18に示す領域24aについてのみ、

$p(x,y)=M3(1,j)u(x,y)+M3(1,i)ref(x+M/b1x,y+M/b1y)$
なる重み付け演算を行なう。次にステップS2011において、同様に図18に示す領域24bについてのみ、

$p(x,y)=p(x,y)+M3(1,j)u(x,y)=p(x,y)+M3(1,i)ref(x+M/b2x,y+M/b2y)$

なる重み付け演算を行なう。尚、上述したように、右辺の $p(x,y)$ に対する加算処理により、左辺の $p(x,y)$ が新たに得られることを示す。そしてステップS2012において、図19に示す領域25aについてのみ、

$p(x,y)=p(x,y)+M5(1,j)l(x,y)=p(x,y)+M5(1,i)ref(x+M/b3x,y+M/b3y)$

なる重み付け演算を行なう。そしてステップS2013において、図19に示す領域25bについてのみ、

$p(x,y)=p(x,y)+M5(1,j)l(x,y)=p(x,y)+M5(1,i)ref(x+M/b4x,y+M/b4y)$

なる重み付け演算を行なう。次にステップS2014において、図19に示す領域25cについてのみ、

$p(x,y)=p(x,y)+M6(1,j)r(x,y)=p(x,y)+M6(1,i)ref(x+M/b5x,y+M/b5y)$

なる重み付け演算を行なう。次にステップS2015において、図19に示す領域25dについてのみ、

$p(x,y)=M6(1,j)r(x,y)=p(x,y)+M6(1,i)ref(x+M/b6x,y+M/b6y)$

なる重み付け演算を行なう。そして最後にステップS2016において、着目マクロブロックにおける図18に示す領域24a、24b、及び図19に示す領域25a、25b、25c、25dに対して、

$p(x,y)=(p(x,y)+M2(1,i)q(x,y)+4)/8$
 $=(p(x,y)+M2(1,j)ref(x+M/1x,y+M/1y)+4)/8$

を演算し、着目マクロブロックの残りの領域には、

$p(x,y)=q(x,y)=ref(x+M/1x,y+M/1y)$

のコピー処理を行なう。

【0053】尚、上述したステップS2002～S2005の場合と同様、ステップS2010～S2016の処理についても、その順番を入れ替えることが可能である。

【0054】以上説明したように本実施形態によれば、マクロブロック単位によるオーバーラップ動き補償を行なうことにより、アクセスする画素数を減少し、加算演算の回数を減らすことができる。従ってオーバーラップ動き補償処理を高速化することが可能となる。

【0055】＜第2実施形態＞以下、本発明に係る第2

実施形態について説明する。尚、第2実施形態において、着目マクロブロックに隣接するマクロブロックはそれぞれ1つの動きベクトルしか持たないものと仮定する。尚、4つの動きベクトルを持つマクロブロックが隣接する場合には、上述した第1実施形態と同様に場合分けを行ない、それぞれに応じた処理を実行すれば良い。

【0056】第2実施形態においては、上述した第1実施形態で示したマクロブロック単位によるオーバーラップ動き補償処理を、画像の上部境界のマクロブロックに適用したことを特徴とする。

【0057】画像の上部境界に位置する着目ブロックにおいては、上に隣接するブロックが存在しないために、該上に隣接するブロックに対応する動きベクトルは、着目ブロックの動きベクトルで置き換えて処理される。即ち、第1実施形態で示した(6)式において、 $u(i, j)$ を $q(i, j)$ に置き換えることにより、演算を行なう。 *

$$\begin{aligned} p(x, y) &= (q(x, y)M2(i, j) + q(x, y)M3(i, j) + l(x, y)M5(i, j) \\ &\quad + r(x, y)M5(i, j) + 4) / 8 \\ &= (q(x, y)(M2(i, j) + M3(i, j)) + l(x, y)M5(i, j) \\ &\quad + r(x, y)M5(i, j) + 4) / 8 \end{aligned}$$

ここで(7)式によれば、図20に示す $M2(i, j) + M3(i, j)$ の1/2を占める。係数が8である要素については、 $p(x, y) = q(x, y)$ となるので、計算を行なう必要がなくなる。即ち、実際に演算処理を実行しなければならない画素が、全体の1/2まで減少することが分かる。

【0061】以上説明したように本実施形態によれば、画像の上部境界にあるマクロブロックのオーバーラップ動き補償処理をさらに高速化することができる。

【0062】例えば、符号化対象である画像のフォーマットがテレビジョンの共通中間フォーマットであるQCIF (Quarter Common Intermediate Format)であった場合、画像全体のマクロブロックの数は11×9個であり、そのうち11個のマクロブロックが画像の上部境界に相当する。従って、画像全体の1/9の領域において、第2実施形態による効果が得られる。

【0063】さらに、画像フォーマットがサブQCIFであった場合、画像全体のマクロブロックの数は8×6個であり、そのうち8個のマクロブロックが画像の上部境界にあるため、画像全体の1/6の領域において、第2実施形態による効果が得られる。

【0064】＜第3実施形態＞以下、本発明に係る第3実施形態について説明する。尚、第3実施形態において、着目マクロブロックに隣接するマクロブロックはそれぞれ1つの動きベクトルしか持たないものと仮定する。尚、4つの動きベクトルを持つマクロブロックが隣接する場合には、上述した第1実施形態と同様に場合分※

$$\begin{aligned} p(x, y) &= (q(x, y)M2(i, j) + u(x, y)M3(i, j) + q(x, y)M5(i, j) \\ &\quad + r(x, y)M5(i, j) + 4) / 8 \\ &= (q(x, y)(M2(i, j) + M5(i, j)) + u(x, y)M3(i, j) \\ &\quad + r(x, y)M5(i, j) + 4) / 8 \end{aligned}$$

*【0058】従って、図9に示す重み係数マトリクス $M2(i, j)$ と図10に示す重み係数マトリクス $M3(i, j)$ の加算結果である、図20に示す重み係数マトリクスを用いることにより、マクロブロック単位のオーバーラップ動き補償処理を行なうことができる。即ち、着目マクロブロックの動きベクトルを用いて予測した画像データに対しては図20に示す重み係数マトリクスを乗じ、左右に隣接するブロックの動きベクトルを用いて予測した画像データに対しては、第1実施形態において図11、図12に示した重み係数マトリクスを乗じる。そして、それらを足し合わせてさらに4を加えた後に、8で割った値を予測値(画像)とする。

【0059】上記演算を数式によって表現すると、下記に示す(7)式が得られる。

【0060】

…(7)

※けを行ない、それぞれに応じた処理を実行すれば良い。

【0065】第3実施形態においては、上述した第1実施形態で示したマクロブロック単位によるオーバーラップ動き補償処理を、画像の左境界及び右境界のマクロブロックに適用したことを特徴とする。

【0066】画像の左境界に位置する着目ブロックにおいては、左に隣接するブロックが存在せず、また、画像の右境界に位置する着目ブロックにおいては、右に隣接するブロックが存在しない。従って、これら存在しない隣接ブロックに対応する動きベクトルは、着目ブロックの動きベクトルで置き換えて処理される。即ち、第1実施形態で示した(6)式において、画像の左境界では $l(i, j)$ を $q(i, j)$ に置き換え、右境界では $r(i, j)$ を $q(i, j)$ に置き換えることにより、演算を行なう。

【0067】従って左境界においては、図9に示す重み係数マトリクス $M2(i, j)$ と図11に示す重み係数マトリクス $M5(i, j)$ の加算結果である、図21に示す重み係数マトリクスを用いて、また右境界においては、図9に示す重み係数マトリクス $M2(i, j)$ と図12に示す重み係数マトリクス $M5(i, j)$ の加算結果である、図22に示す重み係数マトリクスを用いることにより、マクロブロック単位のオーバーラップ動き補償処理を行なうことができる。

【0068】第3実施形態における左境界及び右境界のオーバーラップ動き補償の演算を数式によって表現すると、下記に示す(8)式及び(9)式が得られる。

【0069】

…(8)

$$\begin{aligned}
 p(x,y) &= (q(x,y)M2(i,j) + u(x,y)M3(i,j) + l(x,y)M5(i,j) \\
 &\quad + q(x,y)M5(i,j) + 4)/8 \\
 &= (q(x,y)(M2(i,j) + M5(i,j)) + u(x,y)M3(i,j) \\
 &\quad + l(x,y)M5(i,j) + 4)/8
 \end{aligned}$$

ここで、例えば(8)式によれば、図21に示す $M2(i,j) + M5(i,j)$ の9/16を占める。係数が8である要素については、 $p(x,y) = q(x,y)$ となるので、計算を行なう必要がなくなる。即ち、実際に演算処理を実行しなければならない画素が、全体の7/16まで減少することが分かる。また(9)

式においても、同様に処理すべき画素数が全体の7/16となる。

【0070】以上説明したように本実施形態によれば、画像の左境界及び右境界にあるマクロブロックのオーバーラップ動き補償処理をさらに高速化することができる。

【0071】例えば、画像のフォーマットがQCIFであった場合、画像全体の11×9個のマクロブロックのうち、9個が画像の左境界に、また9個が右境界に存在する。従って、画像全体の2/11の領域において、第3実施形態による効果が得られる。

【0072】さらに、画像フォーマットがサブQCIFであった場合、画像全体のマクロブロックの数は8×6個であり、そのうち12個のマクロブロックが画像の左又は右境界にあるため、画像全体の1/4の領域において、第3実施形態による効果が得られる。

【0073】＜第4実施形態＞以下、本発明に係る第4実施形態について説明する。

【0074】上述した第1乃至第3実施形態によれば、オーバーラップ動き補償の処理速度を向上させることができる。第4実施形態においては、上記各実施形態において使用される重み係数マトリクスの格納領域を節約することを特徴とする。

【0075】上記各実施形態においては、オーバーラップ動き補償をマクロブロック単位で行なうために各重み係数マトリクスは16×16のサイズとなっているが、該マトリクスにおいて、係数が0または8である部分については実際の処理を行なう必要がないため、このような係数は保持しておく必要がない。

【0076】例えば、図10、図11、図12に示す重み係数マトリクスについては、係数が0でない、16×4または4×16の有為係数部分のみを保持すれば良い。また、図9に示す重み係数マトリクスについては、係数が8でない、4×16の有為係数部分を2つと8×4の有為係数部分を1つ保持すれば良い。また図20に示す重み係数マトリクスについては、係数が8でない、4×16の有為係数部分を2つ保持すれば良い。また、図21、図22に示す重み係数マトリクスについては、係数が8でない、16×4の有為係数部分を1つと4×12の有為係数部分を1つ保持すれば良い。

【0077】また、図9～図12、及び図20に示す重み係

…(9)

数マトリクスは上下又は左右に対称性があるため、さらにその半分の有為係数のみを保持すれば良い。

【0078】以上説明したように第4実施形態によれば、上述した第1乃至第3実施形態におけるオーバーラップ動き補償を行なう際に、参照する重み係数マトリクスの最低限必要な係数のみを保持することにより、そのメモリ容量を節約することができる。

【0079】＜他の実施形態＞なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0080】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0081】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0082】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0083】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0084】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになる。

【0085】

【発明の効果】以上説明したように本発明によれば、動きベクトル予測に伴って生じる隣接ブロック間の不連続性の歪を軽減するオーバーラップ動き補償が高速に可能となる。

【図面の簡単な説明】

【図1】本発明に係る一実施形態の動き補償予測符号化における動きベクトル情報と予測画像データの関係を示す図。

【図2】本実施形態における着目ブロックに対する重み係数マトリクス例を示す図。

【図3】上に隣接するブロックに対する重み係数マトリクス例を示す図。

【図4】下に隣接するブロックに対する重み係数マトリクス例を示す図。

【図5】左に隣接するブロックに対する重み係数マトリクス例を示す図。

【図6】右に隣接するブロックに対する重み係数マトリクス例を示す図。

【図7】本実施形態におけるマクロブロックの構成を示す図。

【図8】マクロブロックを構成する各ブロック毎の重み係数マトリクス例を示す図。

【図9】本実施形態における着目マクロブロックに対する16×16の重み係数マトリクス例を示す図。

【図10】上に隣接するブロックに対する15×16の重み係数マトリクス例を示す図。

*【図11】左に隣接するブロックに対する15×16の重み係数マトリクス例を示す図。

【図12】右に隣接するブロックに対する15×16の重み係数マトリクス例を示す図。

【図13】本実施形態におけるオーバーラップ動き補償処理の概要を示すフローチャート。

【図14】本実施形態におけるマクロブロック単位のオーバーラップ動き補償処理の詳細を示すフローチャート。

【図15】マクロブロック内の重み付け領域を示す図。

【図16】マクロブロック内の重み付け領域を示す図。

【図17】着目マクロブロックと隣接するブロックの関係を示す図。

【図18】マクロブロック内の重み付け領域を示す図。

【図19】マクロブロック内の重み付け領域を示す図。

【図20】本発明に係る第2実施形態における重み係数マトリクス例を示す図。

【図21】本発明に係る第3実施形態における重み係数マトリクス例を示す図。

【図22】本発明に係る第3実施形態における重み係数マトリクス例を示す図、である。

【符号の説明】

101 処理対象フレーム画像

102 着目ブロック

103 動きベクトル

104 参照画像

105 予測画像データ

【図2】

4	5	5	5	5	5	5	4
5	5	5	5	5	5	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4

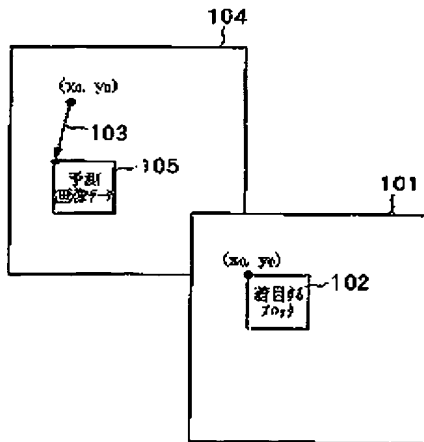
【図3】

2	2	2	2	2	2	2	2
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

【図4】

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
2	2	2	2	2	2	2	2

【図1】



【図5】

2	1	1	1	0	0	0	0	0	0	0	0	1	1	1	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	1	1	1	0	0	0	0	0	0	0	0	1	1	1	2

【図6】

【図8】

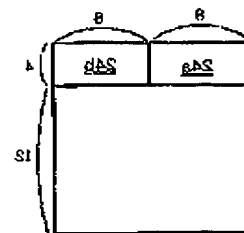
4	5	5	5	8	8	8	6
5	5	5	5	8	8	7	7
5	5	6	6	7	7	7	7
5	5	6	8	7	7	7	7
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	7	7	7	8	8	8	8

(a)

8	8	8	8	5	5	5	4
7	7	6	6	5	5	5	5
7	7	7	7	6	6	5	5
7	7	7	7	6	6	5	5
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	7	6

(b)

【図18】



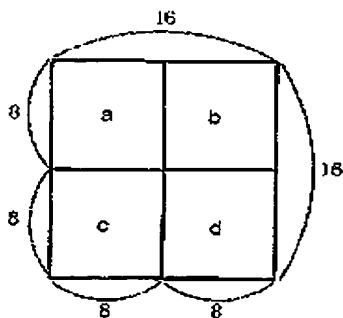
6	7	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	7	7	7	8	8	8	8

(c)

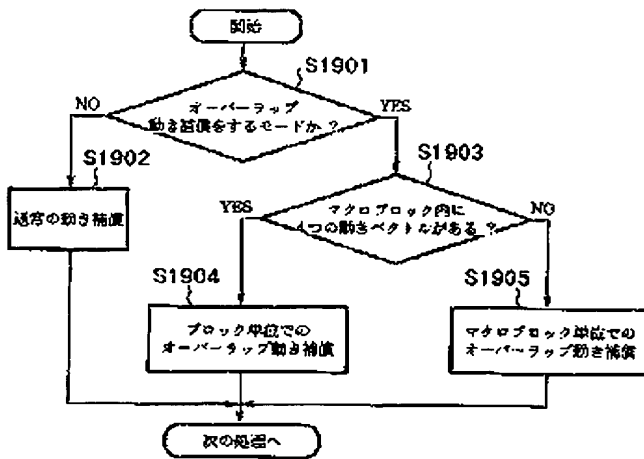
8	8	8	8	7	7	7	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	7	6

(d)

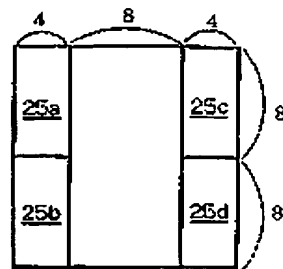
【図7】



【图 13】



【图 19】



【图20】

[illegible]

【图22】

[illegible]

```

graph TD
    Start([開始]) --> S2001{S2001  
上, 左, 右に隣接する  
マクロブロックの動きベクトルの  
数はすべて1?}
    S2001 -- YES --> S2002[S2002  
領域21aの重み  
付け処理]
    S2002 --> S2003[S2003  
領域22aの重み  
付け処理]
    S2003 --> S2004[S2004  
領域22bの重み  
付け処理]
    S2004 --> S2005[S2005  
・領域21a, 22a, 22bに  
おける重み付け加算処理  
・その他の領域における画像  
データのコピー]
    S2005 --> End([終了])
    S2001 -- NO --> S2010[S2010  
領域24aの重み付け処理]
    S2010 --> S2011[S2011  
領域24bの重み付け処理]
    S2011 --> S2012[S2012  
領域25aの重み付け処理]
    S2012 --> S2013[S2013  
領域25bの重み付け処理]
    S2013 --> S2014[S2014  
領域25cの重み付け処理]
    S2014 --> S2015[S2015  
領域25dの重み付け処理]
    S2015 --> S2016[S2016  
・領域24a, 24b, 25a, 25b,  
25c, 25dにおける重み  
付け加算処理  
・その他の領域における画像  
データのコピー]
    S2016 --> End
  
```

[illegible]